For your College Management System, here’s a structured approach to designing each tier. You can follow these instructions in Lucidchart (or any other design tool) to create the necessary diagrams.

1. Front End - Wireframes for Each Module

Create wireframes for each module, including key forms and menu options.

Sample Modules and Fields

1. Login Page
- Form Fields:
- Username
- Password
- Button: Log In
2. Dashboard (Admin, Faculty, Student)
- Admin Menu Options: Student Management, Faculty Management, Course Scheduling, Enrollment Tracking
- Faculty Menu Options: Courses, Enrolled Students
- Student Menu Options: My Courses, Enroll in Course
3. Student Management (Admin)
- Form Fields:
- Student ID
- Name
- Email
- Course Enrollment
- Buttons: Add Student, Update, Delete
4. Course Scheduling (Admin)
- Form Fields:
- Course Name
- Course Code
- Faculty Assigned
- Schedule
- Buttons: Add Course, Update, Delete

2. Controller - Business Logic (RESTful API Endpoints)

Map each front-end form field or action to a corresponding RESTful service in the controller tier. Here’s a layout you can create:

Sample REST API Services

1. Authentication

- POST /api/auth/login: Handles login, validates username and password, and returns JWT token.
- POST /api/auth/logout: Logs the user out.

2. Student Management (Admin)

- GET /api/students: Fetches all student records.
- POST /api/students: Adds a new student to the system.
- PUT /api/students/{id}: Updates a student’s information.
- DELETE /api/students/{id}: Deletes a student’s record.

3. Faculty Management (Admin)

- GET /api/faculty: Lists all faculty members.
- POST /api/faculty: Adds a new faculty member.
- PUT /api/faculty/{id}: Updates faculty information.
- DELETE /api/faculty/{id}: Removes a faculty record.

4. Course Scheduling

- GET /api/courses: Lists all available courses.
- POST /api/courses: Adds a new course.
- PUT /api/courses/{id}: Updates course details.
- DELETE /api/courses/{id}: Deletes a course.

5. Enrollment Management

- POST /api/enrollment: Enrolls a student in a course.
- DELETE /api/enrollment: Unenrolls a student from a course.

3. Model - Database Tier (Schema Diagram)

Design the schema for the MySQL database with tables and relationships.

Suggested Tables and Relationships

- Users Table:
- Columns: user_id (PK), username, password_hash, role (e.g., admin, faculty, student)
- Students Table:
- Columns: student_id (PK), name, email, phone
- Relationships: user_id (FK to Users)
- Faculty Table:
- Columns: faculty_id (PK), name, department, email
- Relationships: user_id (FK to Users)
- Courses Table:
- Columns: course_id (PK), name, code, faculty_id (FK to Faculty), schedule
- Enrollments Table:
- Columns: enrollment_id (PK), student_id (FK to Students), course_id (FK to